Beta Lessons Learned

MODIS SDST Science S/W Transfer Group (SSTG)

Science Advisory Panel Meeting

October 24-25, 1995

S/W Transfer Lessons-Requirements

- The GSFC DAAC requires format descriptions for all files in the science code. To standardize the presentation of the descriptions, a set of SDST-provided templates would be useful.

- SSI&T requirements, e.g. those for test data and metadata, were not enumerated and frozen early enough in Beta to allow for full compliance by the SCFs and SDST.

- SCFs need to place more emphasis on documentation of their deliveries, including system-level documentation, which goes beyond the bare minimum required by ECS. SDST should define the extra documentation.

- UNIX tools for code standards checking (ANSI option of cc, ftncheck) cannot be customized.

- An automated checker in V1 could reduce manual code checks while allowing customization to the ESDIS standards set.

- Beta identified the need for numerous utilities in V1 and V2, e.g. HDF analysis, product QA, and ancillary data.

S/W Transfer Lessons - Design

- ECS release of SDP TK V5 (esp. metadata features) and HDF prototype came too late to help in Beta. Not enough detail was given by ECS on their metadata approaches at an early enough stage to be useful in design of the MODIS API.

- L1B format: hyperslab reads, including all bands at once for each scan line, were much more efficient than single band reads because the Beta format is band by pixel interleaved. However, benchmark tests proved that the Beta format is not optimized for read access, and a reordering of arrays to a band by scan line structure would speed access considerably.

- Standard gridded ancillary products (e.g. NMC) are too coarse for providing accurate values at satellite pixels. Science software developers will need to provide routines which interpolate gridded ancillary data to satellite pixels.

S/W Transfer Lessons - Coding

- Templates or wrappers for SDP tools and L1B reading are time savers and ease code maintenance.

- A 5-10 day code turnaround cycle was often difficult in Beta. 15-25 calendar days is more practical when a large number of changes are needed in a given incremental delivery.

- Final code walkthroughs should not only check for adherence to standards, but should include provision of test output, an HDF product spec., and updated bubble charts. The walkthrough forms have already been modified to include the new items.

- It is most efficient to have the CM receive only "final" deliveries and for the SSTG to manage incremental deliveries.

- SDP TK automated response system allows up to 30 days for non-installation questions. In some cases this impacts code turnaround time. An urgency ranking and more timely feedback is needed for SDP TK usage problems.

S/W Transfer Lessons - Coding (cont.)

- It is necessary to check for temporal matches between ancillary data and corresponding simulated data before processing proceeds.

- The MODIS oceans code uses some approaches unique to the overall set of deliveries: pmake; combined Ratfor, FORTRAN 77 and C; HDF I/O interfaces which are constrained by heritage code (SeaWiFS). In some cases this makes maintenance relatively difficult.

S/W Transfer Lessons - Programmer

- The SAIC Common Approach to S/W Development, including PDL and code walkthroughs, was essential in catching errors early on in development.

- A single makefile can be used to test code on SGI Challenge (modis-xl) and SGI Power Challenge (modispc), if the "-32" compile option is used.

- All Vdata rows in an HDF must be written with one call; multiple calls, one per field in the row, are not supported.

- The rank must be initialized before calling M-API's GMARDM, which retrieves an array's dimensionality information using the FORTRAN interface.

- The SGI cache design necessitates reading data as "slabs," optimizing block sizes to minimize read/write cache conflicts, and ordering C arrays as (lines,bands,pixels) rather than (lines,pixels,bands).

S/W Transfer Lessons - Programmer (cont.)

- Use of reserved system function names (e.g. open() and close()) can cause Sentinel to abort program execution. Routines which open and close files should be renamed (e.g. open_files(), close_files()).

- A Software Development Folder (SDF) for each MODIS process and M-API routine greatly improves software maintenance. A set of SDFs is now in place for land, ocean, atmosphere, L1B and M-API.

S/W Transfer Lessons - Test

- The Beta simulated L1B data set is inadequate for oceans testing.

- Having knowledge of the tests performed on science code at the SCFs facilitates the testing done at the TLCF.

S/W Transfer Lessons - Communications

- More frequent technical interactions are needed between GSC/RDC and Hughes ECS, SDP TK, and EOS-HDF developers.

- More frequent technical interactions are needed between SSTG and SCF programmers and scientists. Possibilities include a listserv, newsletter, augmented WWW site, use of gsfc.modis newsgroup, regular telecons. Interactions as a group, not just SCFs individually with SDST, would ensure more consistency in knowledge base across SCFs.

- SDST needs to provide, early in V1, a single reference guide for SCF developers to include code standards/guidelines, relevant SDST processes (e.g. HDF spec. updating, code delivery), required elements of a delivery, SDST schedules.

- The MODIS anonymous FTP site must be under CM control. This includes directories for M-API release, ICD/HDF product specs., utility software, and data sets.

S/W Transfer Lessons - Comm. (cont.)

- An equivalent to the MODLAND concept of a "Science Liaison" to SDST is beneficial for atmospheres and oceans.

- MODLAND meetings provided quantum leaps in progress and should be duplicated by the atmospheres and oceans groups.

- Face-to-face meetings with SCFs where appropriate, (e.g. the July 1995 visit to Miami oceans SCF) can promote rapid progress.

- Telephone contacts generally resulted in quicker response than e-mail and should be used more liberally during V1, particularly for routine interactions relating to the details of code implementation. Many SCFs are feeling overwhelmed with volume of e-mail they receive. E-mail is still preferred for “official” correspondence regarding delivery dates, approaches, and schedules.

S/W Transfer Lessons - Schedules

- Schedules for new software development (e.g. M-API) must include adequate time for unit and system testing, as well as QA functions (e.g. walkthroughs).

- The SCFs should continue to create and maintain schedules for their code deliveries, synchronized with SDST internal schedules. They should report deviations from established schedules to Ed Masuoka. The work done by the MODLAND group during Beta is a notable success in this area.

- The implementation of SDST schedules was essential to providing a focus for Beta. The schedules need to be updated weekly to be most useful. A single master schedule, with detailed line item descriptions, allows for creation of customized schedule subsets. The master schedule also eliminated redundancies present in the original set of individual discipline schedules.

- Lower-level detailed schedules (e.g. oceans) were necessary supplements to the master schedule for describing the timing of implementing approaches and interactions with SCFs.

S/W Transfer Lessons - Systems

- Sentinel (memory leak checker) has proven useful but is still not fully debugged (installation problems and did not work with certain processes).

- All transfer of large files (e.g. SCF deliveries) should be done over the FDDI network, not Ethernet. When FDDI was down, transfer of oceans delivery took 10-20 times longer over Ethernet.

- Storage capacity for test, simulated, and ancillary data sets on the TLCF is limited. SSTG also needs to spend more time forecasting storage needs for upcoming code deliveries.

- It was beneficial to have a single point of contact within SDST to deal with requests for new accounts and configuration changes by the LTP systems group.

- On-line documents, e.g. Iris Insight, can answer many UNIX questions.